## Foundations of Artificial Intelligence

Prof. Dr. B. Nebel, Prof. Dr. W. Burgard — University of Freiburg
Dr. A. Kleiner, R. Mattmüller — Department of Computer Science
Summer Term 2008

# Exercise Sheet 3

**Due: Friday, May 23, 2008**

**Exercise 3.1** (Programming assignment: Uninformed search)

Download the sources of the search demo applet from the lecture website.[1] Complete the algorithms for depth first search (*DepthFirstSearchFunction.java*) and breadth first search (*BreadthFirstSearchFunction.java*). Please hand in hardcopies of the two modified files.

**Exercise 3.2** (Programming assignment: A* search)

Complete the A* algorithm by extending the file *GridHeuristicOrderingFunction.java* with a reasonable heuristic function in the method *getHCost(SearchNode)*. Please hand in a hardcopy of the completed file.

**Exercise 3.3** (Programming assignment: Hillclimbing search)

In a TSP[2], $n$ cities $1, \ldots, n$ with distances $c_{ij}$ are given, where $c_{ii} = 0$ and $c_{ij} = c_{ji}$ for all $i, j = 1, \ldots, n$. A tour $\pi$ is a permutation $\pi(1), \ldots, \pi(n)$ of the cities. The cost of a tour is $c(\pi) = \sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)} + c_{\pi(n)\pi(1)}$.

(a) Write a problem generator for TSP instances where cities are represented by random points $(x, y)$ in the unit square $[0, 1] \times [0, 1]$. The distance between two cities is defined as the Euclidean distance between the corresponding points.

(b) Implement a hillclimbing algorithm for TSP instances starting with a random initial tour. One step in the search space consists of swapping two successive cities in the current tour. Test and evaluate the algorithm using randomly generated problem instances.

**Exercise 3.4** (Constraint Satisfaction Problems)

Show how a single ternary constraint can be turned into three binary constraints by using an auxiliary variable. You may assume finite domains. How can this transformation be generalized to constraints with more than three variables?
*Hint:* Introduce an auxiliary variable that takes on values which are pairs of other values, and use constraints of the form "$X$ is the first component of the pair $Y$".

The exercise sheets may and should be worked on in groups of three (3) students. Please fill the cover sheet[3] and attach it to your solution.

[1] `http://www.informatik.uni-freiburg.de/~ki/teaching/ss08/gki/searchDemo.tar.gz`
[2] `http://en.wikipedia.org/wiki/Travelling_salesman_problem`
[3] `http://www.informatik.uni-freiburg.de/~ki/teaching/ss08/gki/coverSheet-english.pdf`